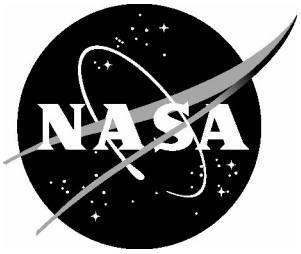
NASA/TM-2003-212654



Launch Vehicle Design and Optimization Methods and Priority for the Advanced Engineering Environment

Lawrence F. Rowell
Swales Aerospace, Hampton, Virginia

John J. Korte
Langley Research Center, Hampton, Virginia

October 2003

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

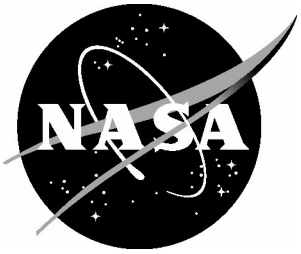
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2003-212654



Launch Vehicle Design and Optimization Methods and Priority for the Advanced Engineering Environment

Lawrence F. Rowell
Swales Aerospace, Hampton, Virginia

John J. Korte
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

October 2003

Acknowledgments

The authors would like to acknowledge Dr. Resit Unal of Old Dominion University in Norfolk, Virginia, for his applications of response surface methods from which the appendix was derived.

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Contents

Acronyms 1
Introduction 2
Three Emphases to Improve Launch Vehicle Conceptual Design 4
  Models 4
  Optimization Methods 5
  Integration Frameworks 6
Launch Vehicle Design Overview 6
Optimization Methods Overview 9
  Parameter Methods 10
  Gradient Methods 10
  Stochastic Methods 12
  Collaborative Optimization 12
Business Use Cases 13
  Vehicle Point Design 14
  Technology Assessment 15
  Uncertainty Analysis 16
Long-Range Vision for Launch Vehicle Design 17
  Automated Multidisciplinary Design Optimization and Concept Ranking 17
  Automated Exploratory Design to Meet Metrics 18
Nonbusiness Use Cases 18
  User Point of View 18
  Verification and Validation (Implementation and Truth Testing) 18
    Report Generation 19
    Configuration Management 19
    Hardware/System Performance and Security 19
Concluding Remarks 19
Appendix 20
References 28

Abstract

NASA's Advanced Engineering Environment (AEE) is a research and development program that will improve collaboration among design engineers for launch vehicle conceptual design and provide the infrastructure (methods and framework) necessary to enable that environment. In this paper, three major technical challenges facing the AEE program are identified, and three specific design problems are selected to demonstrate how advanced methods can improve current design activities. References are made to studies that demonstrate these design problems and methods, and these studies will provide the detailed information and check cases to support incorporation of these methods into the AEE. This paper provides background and terminology for discussing the launch vehicle conceptual design problem so that the diverse AEE user community can participate in prioritizing the AEE development effort.

Acronyms

2nd Gen	Second Generation Reusable Launch Vehicle Program
AEE	Advanced Engineering Environment
APAS	Aerodynamic Preliminary Analysis System
AR	nozzle expansion ratio
BFL	body flap area
BL	ballast weight
CCD	central composite design
CONSIZ	Configuration Sizing
cg	center of gravity
DDT&E	design, development, test, and evaluation
DOE	design of experiment
FR	fineness ratio
GA	genetic algorithm
ISE	Intelligent Synthesis Environment
LH2	liquid hydrogen flow rate fraction
LSA	large-scale application
LVD	launch vehicle design
MDO	multidisciplinary design optimization
MR	mass ratio
NASA	National Aeronautics and Space Administration

OA	orthogonal array
OVAT	one variable at a time
POST	Program to Optimize Simulated Trajectories
RBCC	rocket-based combined-cycle
RSM	response surface method
RSTS	reusable space transportation system
SA	simulated annealing
SMART	Solid Modeling Aerospace Research Tool
SSA	system sensitivity analysis
SSTO	single-stage-to-orbit
SSV	single-stage vehicle
T	thrust
TFA	tip fin area
TPS	thermal protection system
W	weight
WA	wing area ratio

Introduction

For decades, government, industry, and academia have conducted engineering analysis and design of Earth-to-orbit (launch vehicle) system concepts (refs. 1 and 2). The tragic loss of the shuttle *Columbia* and its crew on February 1, 2003 will further generate intensive review of current vehicle systems and future options. Launch vehicle design is a complex, multidisciplinary engineering activity that requires making difficult compromises to achieve balance among competing objectives for the vehicle, including safety, reliability, performance, operability, and cost (ref. 3). As portrayed in figure 1, objectives for safety, reliability, ease of operations, and design margin will increase system weights for both single-stage and two-stage systems. On the other hand, designing for high performance and incorporating appropriate advanced technologies can lead to smaller, lighter systems. Figure 1 also suggests that for some level of maturing technology or tailored performance engineering, single-stage concepts will approach the lower gross weights of two-stage systems. In order to make judgments among many design alternatives, designers need mathematical models that quantify, based on features of the vehicle design, the criteria upon which programmatic decisions will be made. The earlier in the design process that these criteria can be estimated and the compromises understood, the greater is the potential reduction in technical, cost, and schedule risks for the program. For example, it is often approximated in the first order that costs and operations are proportional to vehicle weight and size; however, the effect that some design decisions (e.g., development and application of new technology) will have on costs, operations, or other characteristics are difficult to model (quantify) in sufficient depth to reveal the differences between competing vehicle concepts. Thus, to achieve programmatic goals, such as safety, cost, and schedules, it is necessary to understand and model these relationships and to develop optimization methods that can span the design space to identify the most promising designs. A major NASA program, the Intelligent Synthesis Environment (ISE), was initiated to bring together sufficient resources to design, build, and demonstrate new, powerful, integrated design tools. Several aerospace problems, called large-scale

applications (LSAs), were chosen to be beneficiaries of these tools, and the launch vehicle design problem was referred to as the Reusable Space Transportation System (RSTS) LSA. The ISE program was cancelled in fiscal year 2001, but several activities, including the RSTS application, continued as

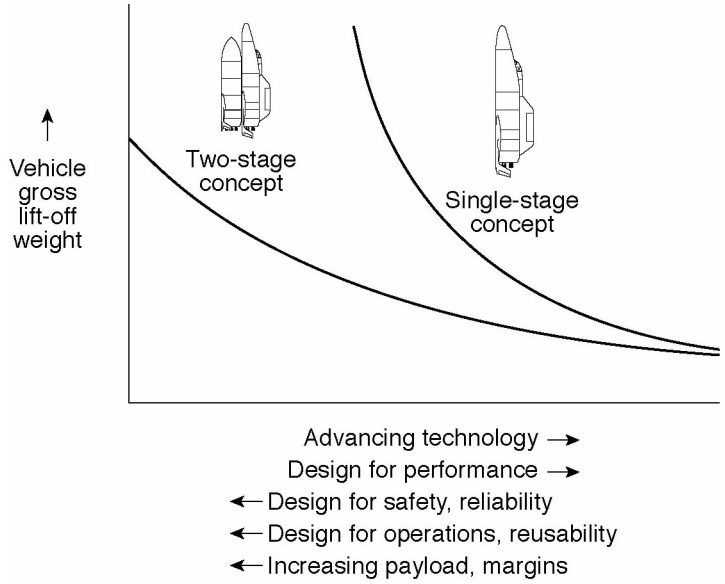


Figure 1. Conflicting design objectives.

focused projects with near-term customers. The former RSTS activity, now pursued as the Advanced Engineering Environment (AEE) program, prioritizes and develops tools and techniques to improve the integration of launch vehicle systems design experts and to improve the conceptual design capability for the Second Generation Reusable Launch Vehicle Program (2nd Gen) (ref. 4). For this reason, AEE is first developing the near-term methods that can be quickly employed to assist the 2nd Gen decision makers in directing their program. The long-term advancement of launch vehicle conceptual design capabilities, also a goal of the AEE program, faces three significant technical challenges:

1. vehicle system definition exists at a very low level of detail during conceptual design, and the relationships among design (vehicle) parameters and design objectives (metrics) are not well understood or modeled;
2. computational methods to optimize models toward selected objectives, i.e., to identify the most promising designs, are very difficult to implement due to the complexity of the codes and the coupling among disciplines; and
3. systems analyses require reliable, timely communication and coordination among diverse engineering experts, computer codes, and data.

The stated goals of the AEE program (private communication, Gary Bollenbacher, "Reusable Space Transportation System (RSTS) Application Requirements Document," June 29, 2001) are to improve distributed group collaboration among design engineers for launch vehicle conceptual design and to provide the infrastructure (methods and framework) necessary to enable that environment. The purpose of this paper is to select three launch vehicle design scenarios, called business use cases, that will be important for the 2nd Gen customer and to propose methods and models necessary to implement these scenarios. AEE users and software developers could elaborate on these examples to identify "build-to" requirements for orderly evolution of the AEE capabilities. The business use cases are at a high level and do not imply any particular launch vehicle concept or technology. A long-range goal for launch vehicle

conceptual designers is to be able to specify in sufficient detail the field of concepts and technologies of interest and then to apply automated analysis and optimization methods that can identify an optimum vehicle design from this field.

The paper is organized as follows. First, a brief overview is given of the challenges facing the AEE team in its efforts to improve conceptual design. Next, overviews of both the launch vehicle design problem and the optimization methods will be given to provide background and terminology for discussing the business use cases. Next, three business use cases, which are launch vehicle design scenarios of current importance to 2nd Gen, are offered as incremental steps in AEE evolution. This section is followed by a long-range vision for a true launch vehicle synthesis capability for the AEE. Finally, some additional, nonbusiness use cases are listed to reflect other challenging aspects of the AEE program.

Three Emphases to Improve Launch Vehicle Conceptual Design

As the complexity and cost of launch vehicle systems have increased, so has the need for significantly improved early systems analysis capability. The types of analyses discussed in this paper address only the early conceptual and preliminary design phases during which the major configuration and technology decisions dictate the largest percentage of total program costs. Advancing the capability for conceptual design to address the problems briefly stated in the introduction requires pursuing at least three major emphases:

1. improvements in the fidelity of disciplinary engineering models and codes, especially in areas of empirical models such as estimations of weights (ref. 5), operations (ref. 6), costs (refs. 7 and 8), and reliability;
2. improvements in computational methods (refs. 9 and 10) that can search the design space and optimize the vehicle system toward selected, weighted objectives; and
3. improvements in software frameworks (ref. 10) that coordinate execution of the coupled engineering disciplines (people and codes) to reduce workload and design cycle time.

Models

Emphasis 1 (improvements in the fidelity of disciplinary engineering models and codes) is primarily an ongoing research activity that relies on disciplinary experts to deduce relationships between design (vehicle) parameters $\{x\}$ and dependent variables $\{u(x)\}$, and to develop realistic models that can be implemented in computer codes. A symbolic problem statement can be expressed as follows.

Let $\{x\}$ = {vector of vehicle design parameters}, $\{u(x)\}$ = {vector of state variables (outputs) from multidisciplinary analyses, $A[x,u] = 0$}, and $\{f_m(x,u)\}$ = {vector of programmatic objectives (metrics)}. Then minimize $Q = Q(w_m f_m(x,u))$ subject to $\{c(x,u)\}$ = {vector of physical and programmatic constraints}, where w_m are weighting factors reflecting the programmatic priorities for this vehicle concept.

The elements in this relationship will be briefly discussed in the sections that follow to provide some insight into the coupling and fidelity characteristics of the disciplinary analyses involved in the launch vehicle conceptual design problem. In addition, these sections will discuss the difficulties in implementing minimization procedures for the system-level launch vehicle design problem and its subproblems (disciplinary codes) due to the characteristics of the disciplines themselves and their software implementations.

Physics-based engineering models, such as those used for aerodynamics, structures, and trajectory analysis, have been employed and validated over decades and can provide a level of fidelity limited primarily by the computational power and design time available. As computing speed increases, conceptual designers have begun to bring many of these higher fidelity, physics-based tools forward to the very early stages of design. However, state-of-the-art modeling in several other launch vehicle design (LVD) disciplines falls short of defining with sufficient fidelity the dependencies between design parameters and vehicle state variables (ref. 3). For example, weights and sizing, operations, cost, safety, and reliability analyses are empirical tools heavily dependent on extrapolation from sparse databases of previous vehicle developments. For vehicle concepts significantly different from past developments or for advanced technologies where hardware has not been developed, such data do not exist. Currently, the resolution from empirical models is far less than physics-based models; thus, statistical approaches, e.g., Monte Carlo simulation, that allow these uncertainties to be modeled are being developed and applied to LVD. Unfortunately, probabilistic uncertainty distributions of component-level characteristics are difficult to determine and validate; yet the primary vehicle metrics $\{f_m(x,u)\}$ (e.g., safety, reliability, cost) proposed for downselecting 2nd Gen vehicle and technology concepts are heavily dependent on empirical models with these deficiencies. (Many other technical problems limit the effectiveness and efficiency of analyses, such as accurate mapping of data across different analysis grids for aerodynamics, heating, and structures programs, but these problems are computational issues rather than issues of understanding the physics.)

As long as this large diversity in fidelity among disciplinary codes exists, the necessity of bringing continually higher fidelity analyses forward into the iterative design process can be questioned (except, perhaps, as used to produce and update simplified, surrogate models for these disciplines). Although progress in incorporating the high-fidelity codes into conceptual LVD is desirable and exciting, it should not preempt a concerted effort to improve methods having less fidelity. New costing tools and databases are being evaluated under the AEE program, and improvement of these tools should be the highest priority for resources dedicated to advancing systems analyses. Inaccurate modeling cannot be corrected by optimization methods or by powerful integration frameworks. Advances in the latter two alone would serve only to create questionable estimates at a faster rate.

Optimization Methods

Emphasis 2 (improvements in computational methods) is also primarily an ongoing research activity for advancement of analytical and numerical optimization methods that are valid for the computer models of interest. The diversity of the disciplinary models used in LVD necessitates that a toolbox of optimization methods (refs. 9 and 10) be available for minimizing $Q(w_m f_m(x,u))$ for the LVD problem. Two major benefits of multidisciplinary design optimization (MDO) are that it achieves higher productivity because feasible solutions are identified more quickly, and it improves understanding of complex engineering problems by revealing and exploiting interactions among the design disciplines that may not be obvious. This challenging area of advancing MDO methods should be the second priority for resources dedicated to improving systems analyses: optimization is key to accelerating exploration of the design space and, thus, provides the greatest opportunity to reduce design cycle time, regardless of modeling accuracy (emphasis 1) or the capability of the software framework (emphasis 3). However, it is important to note that the choice of optimization method significantly affects the options for integration framework used to coordinate the design codes. Some optimization methods can be executed with stand-alone LVD analysis codes (whether an AEE is built or not), but others require that the several codes be integrated or interfaced by some means before the method can be applied.

Integration Frameworks

Emphasis 3 (improvements in software frameworks) requires development of software frameworks that allow the engineers to quickly and easily collaborate to define the design study and execute automated design tasks. The number and characteristics of the analysis tools employed in any particular design effort dictate whether design synthesis and optimization must be conducted manually by the team of disciplinary experts or can be automated via a computational framework. In fact, the existence of a functional framework is necessary in some cases to enable particular optimization approaches.

Engineering models are typically incorporated into automated computational frameworks in one of two ways: 1) as a stand-alone, monolithic, synthesis tool (i.e., compiled as a single executable code) that contains internal modules or subroutines to accomplish each disciplinary analysis; or 2) as a loosely integrated framework in which each discipline is represented by a separate code or codes that exchange necessary data external to the codes. A monolithic synthesis code has the advantage of being fast and executable by a single designer but tends to exclude the disciplinary experts from the design process and can quickly become outdated without continued support and improvement. In addition, these monolithic codes can be difficult to extend or modify.

In contrast, the loosely integrated approach maintains a separate code or codes for each discipline. This form closely resembles the environment typical of many advanced design organizations. These stand-alone tools develop and evolve naturally within their discipline, and each discipline maintains an expertise in their operation. Such tools typically reflect a level of analysis fidelity that exceeds the approximations found in monolithic synthesis codes and are often operated independently by disciplinary experts for single-discipline analysis. In this approach, the computational framework automates the task of executing the contributing codes and exchanging coupling data until the design is converged. The disciplinary experts remain involved to set up and validate initial input files for their disciplinary code, establish ranges of acceptability on key input variables, and suggest alternative solutions for their discipline. Within the bounds specified by the disciplinary experts, this computational framework automates the process of entering input data, executing each of the contributing codes, extracting the required output data, and passing coupling variables on to the next code in the process. For the complex LVD problem, such automation could significantly reduce design time and ensure data consistency between individual disciplines.

The AEE uses this second approach of loosely integrating disciplinary codes, which offers several advantages over using a monolithic synthesis code. For example, this integrated approach enables collaborative design, distributed computing platforms and geographical locations, inclusion of higher fidelity analysis solutions, and it allows disciplinary experts to remain involved. However, this approach has disadvantages as well. Setup and checkout may be tedious, execution can be slower, and, until each contributing disciplinary tool is converted into a design-oriented (noninteractive, batch run) code, they may be highly user-interactive, nonrobust, and poorly suited to calculating gradient information. In addition, codes of commercial origin, where source codes are not available for local modification, present additional challenges and expense, and significant pre- and post-processing (interfacing) is required to automate execution of such codes. Spreadsheet codes also present challenges in automating the calculation of gradient information. Resolving these issues should be the third priority of the AEE activity. The following section briefly describes the major disciplines and characteristics of typical disciplinary codes for launch vehicle conceptual design.

Launch Vehicle Design Overview

Conceptual design refers to systems studies conducted early in the design process and intended to reveal trends and allow relative comparisons among alternatives. Such conceptual design studies provide

quantitative data that can be used by decision makers while the design is still flexible and before the greatest share of life cycle costs are committed. At the beginning of conceptual design, often only the mission requirements (science, defense, commerce, etc.) are known, but, in some cases, additional information regarding vehicle concept, operational approach, and subsystem technologies may also be available. For example, if orbital acquisition with large payload capacity is the primary requirement, such as for space station resupply or satellite deployment, then a rocket vehicle may provide the most cost effective and reliable service to cross the atmosphere into orbit. However, if atmospheric maneuvering or orbital plane changes with small payload are required, such as for some military intercept missions, then an air-breathing concept might offer advantages. These mission requirements flow down, through functional and performance analyses, as derived requirements for the vehicle systems, subsystems, and components determining the many "-abilities" of the vehicle, such as reliability, operability, maintainability, and affordability. In figure 2, the ovals represent design decisions, i.e., the vector of design parameters, $\{x\}$, to be evaluated, and the rectangles represent the analyses $A[x,u]$ that can be conducted to generate the state variables $\{u(x)\}$, depending on the specific issues being addressed.

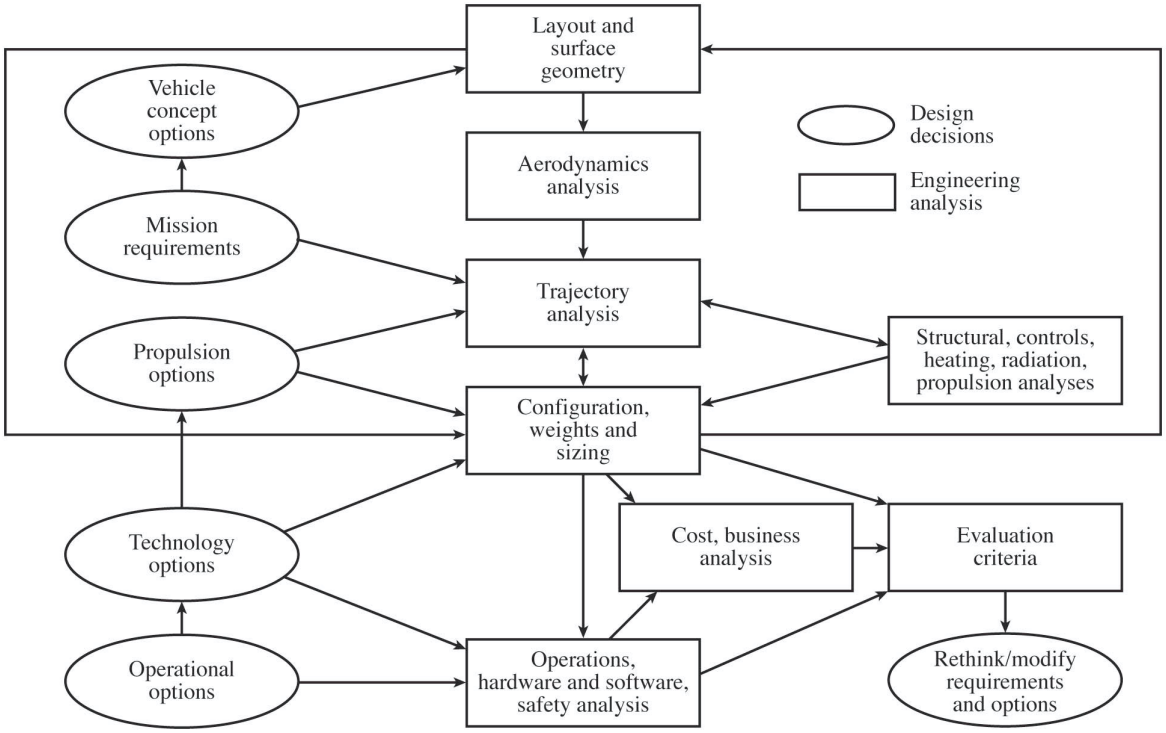


Figure 2. Launch vehicle conceptual design process.

This simplified notion of the LVD process illustrates the variety of disciplines (subproblems) that make up the system-level design problem. The LVD process includes:

- specifying the mission requirements (e.g., payload size, mass, destination, environmental constraints, on-orbit operations, recovery, return)
- selecting a vehicle approach (e.g., rocket or air breather, winged or ballistic, piloted or automated, single or multiple stages, expendable or reusable)
- selecting associated operational scenarios (e.g., assembly, launch, recovery, refurbishment)
- selecting technologies (e.g., structural materials, thermal protection system, avionics, propulsion)

- creating a physical layout and surface geometry that will contain the payload, subsystems, and airborne support equipment
- estimating the ascent and entry aerodynamics (subsonic, transonic, supersonic, hypersonic)
- calculating trajectories and the resulting flight environments
- executing structural, controls, heating, radiation, and propulsion analyses based on the flight environment
- estimating the vehicle weights, dimensions, and center of gravity based on layout, flight environment, and technology selection
- analyzing operations, maintainability, hardware/software requirements, reliability, and safety based on operational scenario, vehicle configuration, and technologies
- estimating life cycle costs (e.g., design, development, test, evaluation (DDT&E); production; operations; disposal) and business performance
- calculating performance and programmatic evaluation criteria (metrics) used to compare alternatives
- using these results to optimize and modify the overall system to better meet mission requirements and design objectives
- continuing this process in an iterative manner to make downselects and deepen the vehicle definition as the concept evolves toward a mature design

Table 1. Characteristics of Representative Launch Vehicle Conceptual Design Codes by Discipline

Discipline	Number and type of variables and constraints	Type of code
Geometry, packaging	Few to many, continuous/discrete	Interactive, graphical, commercial
Aerodynamics	Few, continuous	Interactive, graphical
Trajectory	Many, continuous, dense/sparse	Design-oriented
Weights and sizing	Few to many, continuous/discrete	Design-oriented
Structures	Many, continuous/discrete	Batch, graphical, commercial
Controls	Many, continuous	Batch, commercial
Heating	Few, continuous/discrete	Interactive
Radiation	Few, continuous/discrete	Batch
Propulsion	Few, continuous/discrete	Design-oriented, regression equations
Operations, safety	Many, discrete	Highly interactive, regression equations
Cost, business	Many, continuous/discrete	Highly interactive, regression equations

The conceptual design process is highly coupled (nonhierarchical) and therefore requires significant data exchange and iteration among disciplines and disciplinary codes (engineering models). The diversity of characteristics of the individual disciplinary codes (table 1) requires a variety of optimization approaches capable of treating discrete or continuous design variables, few or many coupling variables and constraints, single solution or multiple minima, and simple or computationally intensive analyses. The number of design variables $\{x\}$ and coupling variables $\{u(x)\}$ present in the full problem can be prohibitively large for analysis and optimization. Thus, the LVD process shown in figure 2 has traditionally been decoupled into four smaller, more manageable MDO problems, as shown in table 2. Two of the four problems address vehicle performance: an *ascent problem*, primarily involving trajectory, weights and sizing, and propulsion analyses, and an *entry problem*, emphasizing geometry, aerodynamics, trajectory, heating, structures, and controls. Once the configuration is designed to meet both ascent and entry performance requirements, a third problem, referred to here as the *economics problem*, may bring the more empirical disciplines, such as operations, software, safety, cost, and business analyses, into the design problem. Since the development of versatile radiation analysis codes (ref. 11), analyses of crew radiation exposure and effects, referred to as the *on-orbit problem,* have been conducted and have demonstrated significant impacts on weights and sizing and safety analyses. Although radiation analyses have typically been treated as post analysis, recent results indicate that they should be incorporated into any process where weights and sizing estimates are important.

Table 2. Typical Approach to Reducing the System-Level Launch Vehicle Design Problem Into Smaller Problems

Ascent problem
Trajectory Weights and sizing Propulsion (Plus aerodynamics and heating if air-breathers)
Entry problem
Geometry Aerodynamics Trajectory Heating Structures Controls
Economics problem
Operations Hardware and software Safety Costs Business (Plus selected other disciplines)
On-orbit problem
Radiation Geometry/layout Weights and sizing Safety

Throughout the LVD process, decision makers use the information from system studies to make choices, i.e., to downselect to smaller and smaller design spaces of concepts and technologies. During this process, MDO methods play an important role by identifying, for each concept, a near-optimum design so that decisions are made based on comparing good designs for every alternative. The following section briefly describes the relationship between the disciplinary code characteristics and the applicability of any MDO method.

Optimization Methods Overview

The evolving AEE framework must implement a versatile set of optimization methods to handle the diverse needs of the individual disciplinary codes given in table 1, as well as to provide methods for the system-level MDO problem. In the past, applications of MDO to the LVD problem were able to address, one at a time, the first two simplified problems (ascent, entry) presented in table 2, but many recent studies (ref. 10) have demonstrated improved vehicle designs by incorporating models from more than one problem area. The characteristics of LVD codes in table 1 dictate what optimization methods are applicable to each discipline, as discussed further below, and clearly, a toolbox of methods must be available. Optimization methods can be broadly classified into three main groups: 1) parameter methods based on design of experiments (DOE) techniques, 2) gradient or calculus-based methods that use derivative calculations, whether by numerical approximation or by code generating techniques like Adifor (ref. 12), and 3) stochastic methods, such as genetic algorithms and simulated annealing. When applying MDO methods to LVD, an implementation strategy is sometimes employed whereby one discipline is chosen as the controlling discipline, or executive (leader), to control execution of the other disciplinary codes (followers) which may themselves be executing optimization processes internally (refs. 13 and 14).

The attributes of each of these three broad classes are briefly described to highlight both their advantages and their difficulties when considering each method for the AEE.

Parameter Methods

The engineering codes used in LVD are typically stand-alone, interactive analyses run by disciplinary experts. Therefore, one approach to system optimization across disciplines is to use methods that do not require the analysis codes to be integrated or automated. Current studies for 2nd Gen could benefit from this approach immediately, even without any AEE framework, whereas other methods, which will be described further, require an advanced framework for implementation. This consideration, and a need to accommodate both discrete and continuous design variables, suggest the application of parameter methods, including response surface methods (RSM), to build polynomial approximation models that represent the relationships between design parameters and design objectives. These approximation models can then be used for MDO and sensitivity analysis. When the RSM approach is employed for MDO, disciplinary experts, each using their independent codes, produce disciplinary-feasible design solutions at several statistically selected combinations of design parameters within the design space, and a polynomial surface is fit to the results. The design points are usually selected by DOE methods such as Taguchi's orthogonal arrays, central composite designs (CCD), and saturated designs. Data generated by the individual engineering codes are collected from the experimental cases and are input to a utility program that calculates the surface approximation equation. Numerical optimization, usually a gradient method, is then performed on these approximation surfaces. The optimal solution point can be some combination of values of the design variables not studied among the originally selected point designs. For a small number of design variables (dimension of $\{x\}$ less than 10), the required number of overall vehicle point designs is usually small, and these designs may be generated by traditional, manual iteration among disciplines and used to generate the vehicle-level response surface equation. An alternate approach is to generate, in parallel, a response surface for each discipline in the problem. The resulting set of discipline-level response surfaces can be implemented as the representative disciplinary modules for subsequent convergence and optimization studies.

Some advantages of parameter methods are: 1) disciplinary analysis integration is not required; thus, interactive, spreadsheet, and commercial codes present no special problem, 2) discrete and continuous design variables may be accommodated, 3) sensitivity information over the entire design space may be inferred from the approximation surface, and 4) constraints can be modified without running additional cases. Also, once the response surface model is validated, the approximation surface can provide a surrogate model for coupling with other applications. Disadvantages of parameter methods are: 1) approximations to some systems may be poor, 2) the approximations yield only a near-optimal solution, 3) as the number of variables grows, this approach becomes unmanageable, and 4) human involvement in designing the experiment and running the experimental cases is required because the stand-alone codes do not allow an automated optimization solution.

Gradient Methods

Gradient, or calculus-based, optimization methods are widely used in individual disciplinary analyses, such as trajectory or structural analysis programs, and in multidisciplinary monolithic synthesis codes where several disciplines are represented within a single program. Gradient methods calculate derivatives of the objective function with respect to design variables, either analytically or by finite differences, to find a path to the minimum solution. Supporting utilities, like Adifor (ref. 12) and AdiC, which generate code to calculate derivatives, are very effective at enabling the use of gradient methods in a broad family of codes. Another technique uses complex variables (ref. 15) to approximate derivatives of real functions. When a function is expanded in a Taylor series with a complex step, the real part is the function value,

and the complex part approximates the derivative. Thus, by evaluating the function at a complex argument, both the function and the derivative are obtained. Reference 16 compares the advantages and disadvantages of these competing approaches for generating derivatives. The use of either automatic differentiation or complex variable approaches in an MDO environment becomes problematic for commercial codes, where source code is not available for modification, and for spreadsheet models.

Because gradient methods typically require many more function calls (i.e., vehicle point designs) than parameter methods to determine a vehicle-level optimum, they must be implemented in a manner to produce converged vehicle information quickly. In the past, the popular approach has been to develop a monolithic synthesis code with the following advantages: 1) a mathematically rigorous optimal solution may be found, 2) large numbers of design and state variables and constraints can be handled, 3) a consistent vehicle model across disciplines may be guaranteed, and 4) once developed, these codes can be executed with little human involvement while they run to completion. Disadvantages of this approach are: 1) these methods do not accommodate discrete variables or nonsmooth design spaces because discontinuous derivatives result, 2) sensitivity information is only known along the solution path, rather than across the design space, 3) the sheer size of these codes may make them impractical as surrogate models for other applications, 4) substantial human effort is often required to merge disciplines into a single code, and 5) many disciplines, because of their complexity or interactive nature, do not lend themselves to integration (see the discussion of system sensitivity analysis that follows).

As noted previously, gradient methods are recommended for use with a monolithic synthesis code in order to achieve reasonable execution speed. However, the monolithic framework itself is responsible for the last three disadvantages listed previously; they are not problems with the gradient method itself. These disadvantages can be avoided by using a subclass of gradient methods, known as system sensitivity analysis (SSA), which does not require explicit analysis integration. Instead, the characteristics of SSA lend themselves well to loosely integrated computing frameworks. For highly coupled design problems, convergence of a design traditionally requires time-consuming internal iteration among disciplines. To reduce optimization time, SSA takes advantage of the Implicit Function Theorem to decompose the larger problem into a set of parallel disciplines that calculate their own gradients. Each discipline simultaneously calculates its local derivatives with respect to design variables and interdisciplinary coupling variables. Local derivatives from all disciplines are combined into a single, linear matrix equation (the global sensitivity equation) that can be solved to simultaneously generate all total derivatives for a numerical optimizer. As a result, no iteration between disciplines is required during gradient calculation. Because the disciplines determine their local derivatives independently and in parallel, this method is also well suited to a design environment in which the design tools have not yet been integrated into a computational framework.

Advantages of the SSA method are: 1) time-saving parallel execution of disciplines during gradient calculation, 2) time savings due to reuse of many of the local sensitivities between optimization steps, and 3) suitability to analysis environments lacking a monolithic synthesis code. Disadvantages include: 1) additional complexity introduced by a matrix solution, 2) possible expansion of individual subproblems due to large numbers of coupling variables, and 3) full analysis (with probable iteration) is still required at fixed point, nongradient solutions.

While parameter and gradient methods described thus far have proven advantageous to conceptual studies, these techniques are not applicable to some classes of problems. For example, optimization of interplanetary trajectories is difficult because the design space is discontinuous with localized minima. Conventional calculus-based and response surface methods are not effective in such domains, and typically, exhaustive grid search methods are employed. For domains of this type, or for problems involving discrete design variables, stochastic methods, such as genetic algorithms or simulated annealing, can be used more efficiently to span the design space.

Stochastic Methods

One of the better-known stochastic methods is genetic algorithms (GAs), which are designed to mimic evolutionary selection. Each individual design candidate is represented by a string that is a coded listing of design parameter values. The string is analogous to a chromosome with genes for the various parameters. The objective function is evaluated for an initial population of candidates, and these candidates compete to contribute new members to future populations. The advantages of GAs are: 1) discontinuous problems, multiple local minima, and discrete variables present no special problem to the method, 2) setup and execution are not difficult, and 3) only analysis function evaluation, not gradients, is required. The disadvantages are: 1) a large number of function evaluations are typically required, 2) only near-optimum solutions are found, 3) no sensitivity information is developed, and 4) to be practical, all disciplines must be integrated into a single code.

The simulated annealing (SA) algorithm, like genetic algorithms, employs randomized search techniques to solve multivariate optimization problems. An analogy to the statistical mechanics of annealing of solids provides the strategy for optimization. SA is a general optimization tool that can be used to solve problems involving both continuous and discrete variables. Like GAs, SA algorithms require a large number of function evaluations to reach an optimal solution. SA algorithms have advantages and disadvantages similar to those listed for GAs above.

Collaborative Optimization

A developing strategy, termed collaborative optimization (ref. 13), allows disciplinary codes to be executed in parallel, while each controls its own design variables and satisfies its own constraints. In this approach, a system-level optimizer minimizes the overall objective and coordinates the subproblem optimizers through negotiated agreement on coupling variables. This strategy models the practical aspects often employed by design teams, but has improved communications and coordination constructs. Collaborative optimization is a design architecture specifically created for large-scale, distributed-analysis applications. This decentralized design strategy allows domain-specific issues to be accommodated by disciplinary analysts, while requiring interdisciplinary decisions to be reached by consensus.

Advantages of the collaborative optimization architecture are: 1) modification of codes or explicit integration into an automated computing framework may not be required, 2) subproblems can be optimized by the best-suited method, 3) subproblems can be added or modified with relative ease, and 4) a large number of variables can be efficiently accommodated. As the number of disciplinary-specific design variables increases and the relative interdisciplinary coupling decreases, the performance of this approach improves.

For each of the approaches described previously, the strategy for communication and control among the optimizer and disciplinary codes was briefly stated; figure 3 (private communication, Dr. John R. Olds, Georgia Institute of Technology, 2002) summarizes the bases for selecting an appropriate MDO method. New MDO methods specifically tailored to complex problems involving LVD disciplines are being demonstrated (refs. 9 and 10), and if the developing AEE can implement these methods, analysis results can be improved and design cycle times reduced. By providing synthesis and optimization capabilities, the AEE will allow designers to quickly explore a design space, evaluate numerous design alternatives, determine their sensitivities, and downselect based on optimized, comparative data. In the next section, three design scenarios are described to serve as examples employing various MDO methods and other developing analysis techniques. These business use cases are provided so that AEE developers and users will have a basis for discussing priorities of various methods and test cases for their validation.

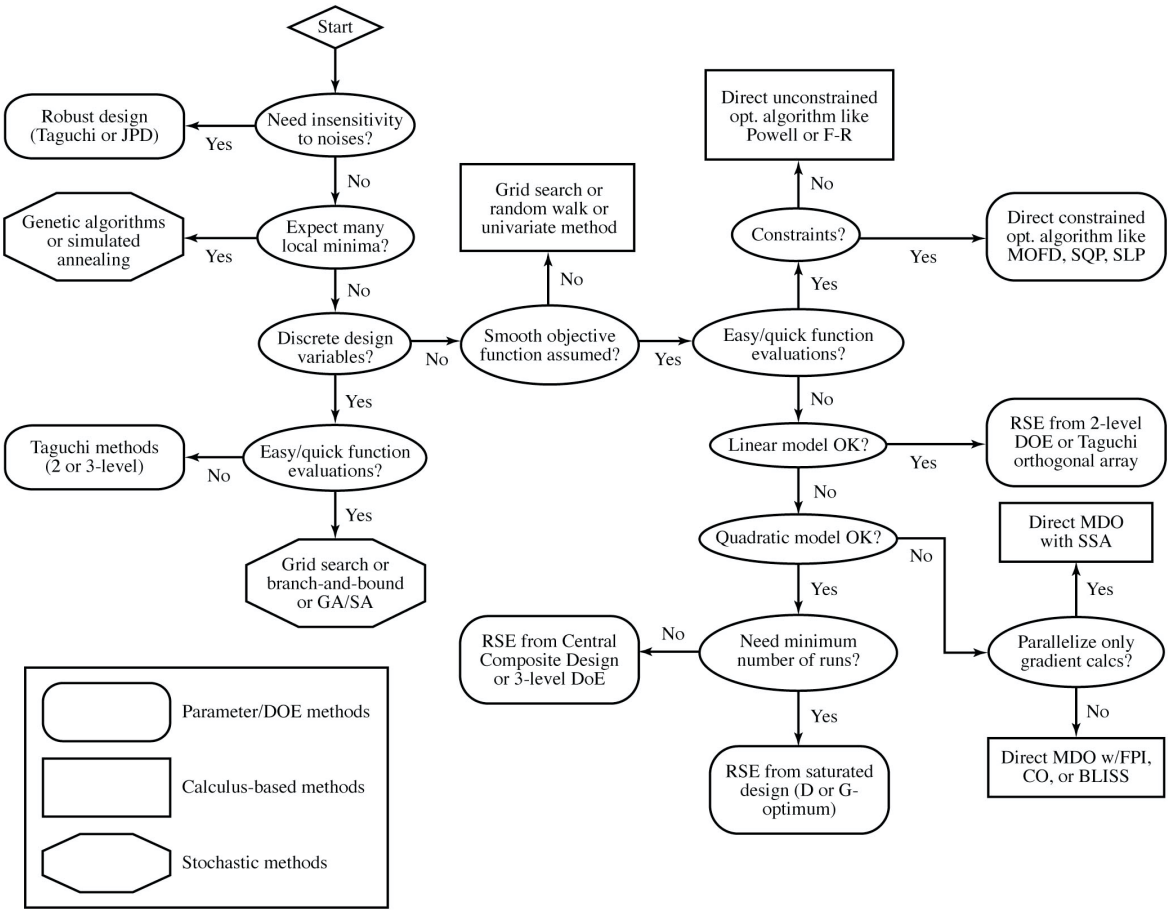


Figure 3. Decision logic for choosing multidisciplinary optimization methods.

Business Use Cases

For the purposes of introducing business use cases for LVD, design and analysis are considered to be different capabilities, although they require the same engineering disciplines. *Analysis* evaluates a given concept in terms of its performance, such as structural, aerodynamic, trajectory, costs, operations, customer metrics, etc. *Design* is the development or synthesis of a configuration in an attempt to meet specified performance metrics. (Architecture assessments look beyond the vehicle elements to the entire infrastructure necessary to support and operate that concept. Similar to vehicle studies, architectures can be analyzed or can be designed to meet performance objectives if the engineering models of the infrastructure exist in sufficient fidelity.) For example, NASA currently uses the analysis framework Recipe (ref. 17) to assist communications among launch vehicle analysis codes that estimate various performance metrics for a concept. Like Recipe, the AEE will begin with analysis, but the evolving capability should move quickly toward design by supporting design-to and optimization capabilities.

In this section, three example design (not analysis) scenarios, or business use cases, will be discussed to illustrate methods that are desirable for the AEE. With the background provided earlier, these design scenarios can be quickly abstracted; we will reference published papers that provide specific details and

results to serve as check cases for evaluating the AEE implementation of that use case. Specific implications on the AEE of each business use case will be identified.

Vehicle Point Design

The measure of suitability for new vehicle concepts, technologies, operational procedures, etc. is most often a comparison of the new concept against another, well understood concept referred to as a baseline design. For such comparisons to have validity, the baseline, and all other designs with which it is compared, must be refined and optimized for the level of definition available so that misleading inferences will not result. In this paper, *vehicle point design* refers to the process of optimizing a specific concept with selected technologies over the flight profile so that it is a good representation of that design.

The design team begins the point design by making specific choices, reflecting agreed-upon ground rules and assumptions, for each design decision as represented by the ovals in figure 2. Obviously, to make fair comparisons, all concepts must use common assumptions, e.g., material strength properties, thermal protection system (TPS) areal weights, propulsion performance, etc. (When concepts are developed by different teams, understanding these assumptions is often the most difficult part of the comparison task. The effort to establish common assumptions can be extensive; however, this matter is less tool-dependent than designer-dependent.) Each disciplinary expert involved will set up and validate his physics-based or statistical models representing that concept and reflecting the chosen technology set. Both the vehicle configuration and the technologies are defined by design parameters that have some range of realistic possibilities. Optimization can then be conducted over this associated range of values to refine the point design and, if sensitivity information is available, a robust design that may be off-optimum can be also be identified. (*Robustness* refers to a concept that is insensitive to adverse changes in the uncertainty variables.) Depending on the computational framework used (i.e., stand-alone codes, monolithic codes, or loosely coupled codes), various MDO methods may be employed to perform the optimization. Examples of applications of MDO for point design follow.

In reference 18, the ascent problem (five nonhierarchically coupled engineering disciplines: trajectory, weights and sizing, heating, aerodynamics, and propulsion; see table 2) was solved for a conical, vertical takeoff, horizontal landing, single-stage-to-orbit (SSTO) vehicle concept with ejector scramjet, rocket-based, combined-cycle (RBCC) engines. The study objectives were twofold: 1) determine the values of three continuous design variables (initial thrust-to-weight ratio between 1.2 and 1.4; rocket mode transition Mach number between 12 and 18; and engine cowl wraparound angle between 180° and 360°) that resulted in the lowest dry-weight vehicle, and 2) given uncertainties in the expected engine weight, airframe weight, and scramjet engine specific impulse, determine the design variable values that resulted in a near-optimal, robust design.

The five disciplinary tools were available as separate, stand-alone, analysis codes, and the required vehicle point designs had to be converged manually among the codes. This fact, and the desire to conduct a robust design, dictated that a parameter method be employed. The three design variables were discretized to three evenly spaced values, and the three noise (uncertainty) variables were each discretized to two values (nominal and degraded). Thirty-nine different vehicle point designs (forming a central composite design experimental array) were generated and used to fit a 22-term, second-order response surface of the design space as a function of the design and noise variables from which the minimum dry-weight point design was identified. The 32 intersecting point designs were used to construct signal-to-noise ratios for each combination of discretized design variable values. Maximizing Taguchi's signal-to-noise ratio (equivalent to finding a design that nearly minimized dry weight but was not overly sensitive to adverse changes in the noise variables) led to identification of a robust point design.

This example demonstrates that parameter methods can produce a substantial amount of information (minimum dry weight solution, robust design solution) without any cultural change for the design team or use of a computing framework, such as the AEE. Also, it suggests that because of the tremendous value to be derived from parameter methods, the AEE should provide capabilities to support their application. To implement a study of this kind, the AEE must: 1) provide tools to capture the experiments (DOE matrix of design parameter combinations), 2) execute the programs in specified sequence while repetitively varying parameters as required by the DOE matrix, 3) communicate disciplinary coupling data among programs, 4) build output data tables and response surfaces, 5) conduct an optimization (typically with gradient methods), and 6) conduct a robust design of the vehicle. This RSM approach to MDO, while not new, offers significant advantages compared with the typical one variable at a time (OVAT) trade study process now in use. The RSM approach exploits interactions among the design disciplines providing greater understanding of disciplinary interactions, and it reduces design cycle time by identifying feasible solutions more quickly. Optimization, in general, can lead to nonobvious solutions, and RSM, in particular, can build expert knowledge (insight) because sensitivity data across the design space are approximated. (The appendix provides a tutorial of RSM and two examples demonstrating applications of the method.)

Because the design variables of reference 18 are continuous, the optimization could also be conducted by gradient or stochastic methods instead of by RSM if the five disciplinary codes could be adequately integrated. Therefore, additional implications of this use case on the AEE would be to 7) automate the execution sequence of disciplinary codes under control of a system-level optimizer (leader) to coordinate the subproblem optimizers and to minimize the overall objective, and 8) provide a toolbox of gradient algorithms and GAs for the subproblems and leader applications. These methods will generally improve the efficiency of the solution compared with RSM because the integrated tools can run without human intervention once setup is done. Point design test cases that could support these additional AEE developments can be found in references 19 (RSM) and 20 (gradient method) describing the optimization of the aerodynamic configuration for a rocket, SSTO vehicle, considering disciplines from both the ascent and descent problems (table 2).

Technology Assessment

Once a point design has been optimized for a particular configuration and technology set, a series of comparisons can be made by introducing replacement technologies in major subsystem areas to determine the advantages or disadvantages they offer to the baseline vehicle metrics (objectives). The modified vehicle designs must be optimized independently for each technology alternative so that a good design and a fair evaluation are produced. However, the goal is not to conduct OVAT technology comparisons, which ignore potential coupling among the subsystem technologies, but to introduce the full array of options to find the combination of technologies that optimizes the vehicle metrics. This optimization problem involves discrete (technology selection) variables and, thus, for those disciplines reflecting the technology options, gradient methods cannot be implemented. MDO may be accomplished by either RSM or GA methods. (These two approaches are not equal: the RSM approach provides an approximation of the metrics across the design space, and does not require integrated codes; GA does not provide an approximation of the metrics and does require integrated codes.) Building on the models developed for the point design study of the previous use case, new disciplinary models must be created for each alternative technology candidate to reflect the impacts (performance, reliability, ground operations, cost, etc.) that the technology has on the vehicle. As in the point design problem, feasible combinations of technologies (discrete variables) are identified, and, depending on the MDO method to be used, either a DOE matrix or a GA initial population can be constructed. A series of point designs (optimizations) will be conducted to complete the process.

In reference 21, two disciplines, costs and weights and sizing, were integrated to evaluate the technology impact of material type (three discrete values: Al, Al-Li, composite) for seven major vehicle structural components that account for the majority of vehicle dry weight: liquid hydrogen tank, liquid oxygen tank, hydrocarbon fuel tank, wing section structure, wing tip fin section structure, basic structure, and secondary structure. The two study objectives were to optimize the vehicle for minimum dry weight and for minimum cost (DDT&E costs as a function of both weight and technology readiness) by determining the best combination of materials for the seven major vehicle sections. The GA method was used because the design problem was characterized by discrete design parameters, and the results yielded different designs (material selection) for the two different objectives. In reference 22, this problem was also studied by using parameter methods.

The primary implications of this use case on the AEE are that: 1) a means must be provided for each discipline expert to create the DOE experimental matrix or GA initial population and to organize a family of models representing the combinations of technologies to be studied, 2) these models must be executable in sequence as called for by the MDO executive, and 3) the technology assessment experimental data must be collected for display and analysis. The price paid for this optimized solution is the potentially large setup time required of the disciplinary experts; however, once setup is accomplished, many studies can be run with these models and the optimized results will present a more realistic picture of the vehicle sensitivities to technology replacement.

In reference 23, disciplines from the ascent and economics problems (table 2) were combined into a single analysis code that was amenable to a gradient-based optimization approach. Optimization was performed to minimize total vehicle development cost, taking into consideration the costs and savings potential of an advanced technology maturation program addressing nine core technologies (main propulsion, composite structure, avionics, tanks, TPS, actuation, prime power, electrical conversion, reaction control system/orbital maneuvering system). In this implementation, the representative DDT&E equations were replaced with a commercial cost-estimating model running on a separate platform and communicating over an internet link. The results from this study indicate that by proper apportionment of technology development funding, vehicle development cost could be reduced by an estimated 15 percent.

This study was implemented as a hybrid architecture in that a commercial code was loosely coupled to a monolithic code, which controlled the optimization. The implications for the AEE, beyond those already identified, are: 1) provide means for communicating with remote applications, and 2) provide optimization leader-follower model constructs with means to ascertain that a mathematically rigorous optimal solution is being generated.

Uncertainty Analysis

Risk analysis is a technique used for a variety of problems that contain significant elements of uncertainty; it can be applied to many aspects of the LVD problem including weight, performance, costs, and schedule estimation. For example, weights estimation of launch vehicle concepts is critical to the business use cases for point design and technology assessment, yet it is fraught with uncertainty. Probabilistic uncertainty distributions of component-level weights are difficult to determine and validate, but, when data are scarce (i.e., uncertainty statistics cannot be based on historical data), expert judgment can be substituted to define the statistical parameters. Thus, using a Monte Carlo simulation approach and expert judgment to define statistics (e.g., mean and standard deviation, or mean plus maximum and minimum values), weight estimates can be produced for various probability distributions and for a specified percentile of cumulative probability. (Sensitivity derivatives have also been used in modeling uncertainty for continuous variables (ref. 24).) Thus, competing vehicle designs can be compared based on weight risk rather than on simple point estimates of weight. In addition, information of this form is

more useful to follow-on cost analyses and to downselect decisions than are simple point-design weight estimates.

In reference 5, application of risk analysis and expert judgment is demonstrated for launch vehicle weights estimation. Of course, the assignment of statistics can be extended beyond component weights to represent predictions of many advanced technology characteristics, such as performance parameters, costs, and schedules. Thus, analogous to the single-discipline weights estimation problem, the technology assessment use case can be extended to reflect the uncertainty inherent in key parameters that define advanced technology candidates; in this way, estimates of vehicle metrics can be provided in terms of confidence intervals. The implications of this use case for the AEE include: 1) the ability to quickly assign statistics to a broad range of characteristics for vehicle components and technologies, 2) execution of Monte Carlo simulations, and 3) presentation of confidence intervals at both the discipline and system levels. The addition of this methodology to the technology assessment use case makes it possible to perform vehicle optimization while considering technology alternatives with uncertainties.

Long-Range Vision for Launch Vehicle Design

This section offers an unconstrained view of what would be desirable for future LVD analysis and optimization, and therefore for the AEE capability. Simply stated, the AEE should, in the long term, enable a fully automated (meaning no interaction by the designer required after problem setup) examination of a range of concepts and technologies. Some scenarios for such a capability are offered below.

Automated Multidisciplinary Design Optimization and Concept Ranking

Many variations on this theme could be proposed, but the eventual goal is to allow experts to define the customer metrics, define which vehicle and technology options (including estimates of uncertainty) will be considered, and then initiate an automated procedure which optimizes the design without further human intervention. This automated procedure would identify feasible designs, calculate customer metrics, rank concepts, generate charts and graphics showing comparisons among these feasible designs, and present comparison and sensitivity data produced during the optimization.

One of the goals of the AEE development is to provide tools to assist vehicle designers to set up and run their disciplinary models, alone or in a multidisciplinary process. This assistance will be one of the most important functions of the AEE, in addition to making available a toolbox of MDO methods, if this long-range business use case is to be approached. To execute this use case, for each design decision (represented by the ovals in fig. 2), the designer must identify all of the options (concepts and technologies) to be considered and then must build the family of disciplinary models reflecting these options. Next, the designer must assign a realistic range of values, with uncertainties, for all parameters required to characterize those alternatives. (Technologies would be defined by both their physical and performance parameters, including best estimates of statistical uncertainties.) Design assumptions and ground rules would be reflected in the selection and definition of the concepts and the parameter values describing the options being considered. The customer metrics and their weights are specified, and the design process is begun. The AEE conducts an automatic synthesis, optimization, technology trade study, and evaluation of all concepts, while reflecting uncertainties, that will satisfy the mission requirements and ranks them according to weighted criteria (metrics) specified by the customer (program office).

Some implications of this scenario are: 1) all disciplinary codes will have a design-oriented (batch run) version and a data dictionary allowing communication of all input and output variables without human interaction, 2) a database of concept and technology alternatives to be considered must exist for each

design decision (represented by ovals in fig. 2), 3) the concepts database must include generalized parametric models (codes) in every discipline set up to represent each concept and technology set needed to analyze that concept, 4) a technology database must exist with a parametric, statistical description of characteristics to allow uncertainty analysis during trade studies, and 5) an optimization procedure, and all MDO and risk methods needed to execute it, must be defined and implemented within the AEE for each combination of disciplines to be executed.

One realistic approach to attempting this long-range business use case would be first to implement in the AEE (i.e., integrate codes for) each of the simplified problems from table 2, and then to couple the four problems via a collaborative architecture into a single, system-level optimization problem. If this design loop can be closed, then not only can traditional design-for-performance objectives be optimized, but also cost-optimized or safety-optimized designs may be generated. Obviously, this capability will evolve through incremental steps that implement the necessary functions a few at a time while delivering useful capabilities at each major milestone.

While considering this long-term vision for LVD, other, very long-term visions also become apparent. If the process described previously does not find feasible designs meeting a particular metric, e.g., cost per pound to orbit, then an extension of this use case can be envisioned.

Automated Exploratory Design to Meet Metrics

To implement this use case would require a method to make minor variations to specified design parameters, outside the initial limits defined by the designer, until some combination (in a least squares sense) of the configuration and/or technology parameters yields a design that satisfies the critical metric. Designers and technologists would then need to evaluate the feasibility of achieving those resulting technology or configuration parameters. This approach would require simple, additional capability from the AEE. A more radical, but interesting approach would be to invert the process of this use case.

Nonbusiness Use Cases

The use cases abstracted in this section are related more to the software, hardware, and administrative aspects of the AEE which are likely to be very challenging exercises for the developers who plan and build the AEE system and infrastructure. These use cases are simply defined in terms of an activity that will be conducted by one of the AEE actors, and each use case must be expanded and detailed early in the AEE design phase.

User Point of View

For each study scenario to be supported by AEE (including the business use case defined previously), single designer, or multiple designers operating from various remote sites, must access, change, test, monitor, and view the design elements of the AEE. These design elements include: a database of concept disciplinary models (analysis code versions); a database of advanced technology candidates; parameter value, range, and uncertainty specification; execution procedures; and results presentation. Also, this point of view must be considered for an automated/ optimization procedure set up by a designer and running without further interaction.

Verification and Validation (Implementation and Truth Testing)

For each of the many disciplinary models brought together to form the design system, test cases must be provided to prove that 1) each model is a fair representation of the physical or statistical system, and 2)

these models are integrated via the correct variables. The suite of test cases established must be sufficient to verify (to confirm that the proper models are installed) and validate (to confirm the system of models represents the intended physical configuration and environment) the set of integrated tools for the study to be executed.

Report Generation

Summary reports will be generated automatically by extracting from the database information describing: input parameters, analysis code versions, execution procedures, parameter ranges explored, design space sensitivities, etc.

Configuration Management

Configuration management must be conducted so that study results can be reproduced, when necessary, from that study's database of input values, custom code versions, execution procedures, and output values, etc. Integrity of the past results and environment must be maintained because new program versions and data are continually generated.

Hardware/System Performance and Security

Expectations for study turnaround time and security must be confirmed by simulating each proposed implementation of the integrated tool set, including most likely hardware (computers, file storage), software (disciplinary codes, databases, interfaces), operating systems, remote facilities, and communication methods.

Concluding Remarks

The first operational version of the Advanced Engineering Environment (AEE) will have an analysis capability equivalent to the current Recipe system, but will also enable evolution into significantly improved design capabilities. Areas of opportunity open to the AEE program for improving the conceptual design capability, in order of relative priority, are: 1) increasing the fidelity of empirical disciplinary models, 2) improving design solutions and insight by implementing design of experiments and optimization methods, and 3) reducing workload and design cycle time through advanced frameworks.

Even without tool integration via AEE or Recipe, the Second Generation Reusable Launch Vehicle Program can benefit from application of design of experiments and response surface methods to the vehicle studies now underway. An operational AEE can provide the framework for additional methods described in this paper and in numerous previous studies, some of which have been identified to serve as examples and test cases to assist the AEE development team. As these methods are incorporated, various simplified multidisciplinary design problems will be implemented, and these simplified problems can, over the long term, evolve into a robust, system-level design capability. Such a capability will allow vehicle designers to rapidly consider a number of fully converged design alternatives in a very short time without sacrificing design detail, thus improving the quality of conceptual space vehicle design.

Appendix

Overview of Response Surface Methods Proposed for Launch Vehicle Design Studies

This appendix provides a tutorial of response surface methods (RSM) that should be introduced into the study process of the Second Generation Reusable Launch Vehicle Program (2nd Gen) even before the Advanced Engineering Environment (AEE) is operational. Included in the appendix are two examples demonstrating applications of the method. Dr. Resit Unal of Old Dominion University in Norfolk, Virginia was instrumental in applying these methods to the launch vehicle design (LVD) problem and in developing the examples below.

Why should response surface methods be applied to the current launch vehicle study activity?

- RSMs can identify an optimum combination of technology or configuration options rather than evaluating one or two options at a time with potentially misleading results.
- It is a useful and timely enhancement to the current process because RSMs require no changes to the analysis codes or the integration framework, i.e., Recipe or AEE.
- The RSM approach requires: 1) an up-front effort to define and execute the experimental cases, and 2) postprocessing of data to construct the polynomial surface to which optimization and robust design methods will be applied. RSM does not require codes to be integrated or loosely coupled.

Background

Launch vehicle systems conceptual design is characterized by:

- Complex, multidisciplinary studies of architecture, configuration, technology, and operational options considering many parameters at many values
- A design approach that is typically experience-based, trial and error variation of parameters where combinations can number in the thousands
- A process that is usually manpower and computationally intensive and may be terminated early due to budget or schedule pressures
- A parametric design space that might not include the optimum

Typical launch vehicle design practice is characterized by:

- New designs based on earlier concepts
- Corrections of problems in performance or reliability with better, costlier components
- A one variable at a time (OVAT) approach rather than parametric design, such as Taguchi method; where opportunities to improve quality without increasing cost are overlooked

The OVAT approach typical of most launch vehicle design studies:

- Produces no information about interactions between variables and disciplines
- Seldom leads to identification of the optimum solution
- Is inefficient and may produce misleading results

Some advantages to robust design methods include:

- Improving efficiency of resources (people, computational) and identifying improved designs whether or not the combination of parameters identified as optimum was tested
- Identifying most important design factors among those studied (weight, cost, etc.)

Launch Vehicle Conceptual Design by Response Surface Methods

Robust design method (design of experiments (DOE) method)

- Systematic, efficient approach to find near-optimum solution for performance characteristics such as cost or weight
- Uses existing analysis codes to quantify design objective function
- Can handle continuous or discrete variables and nonlinearities
- Can produce measures of "noise" or variances
- Used successfully in consumer products and in aerospace field
- Uses orthogonal arrays from DOE theory (for efficient design space sampling) to study large numbers of parameters with small number of experiments

Orthogonal (fractional factorial) arrays compared with full-factorial studies

- Examples: varying 4 parameters at 3 levels requires only 15 experiments, rather than $3^4 = 81$ for full-factorial design
- Varying 8 parameters at 3 levels requires 45 experiments; full-factorial design would require $3^8 = 6{,}561$ experiments
- Varying 13 parameters at 3 levels requires 105 experiments; full-factorial design would require $3^{13} = 1.6$ million experiments
- Optimum may not be (and usually is not) one of the selected experiments
- Uses mathematical model to predict optimum
- Math models enable multidisciplinary integration and fast analysis

As illustrated in figure A1, analysis codes represent the relationship between design variables and the performance metrics to be estimated, and properly selecting experimental cases can lead to efficient optimization studies using RSM.

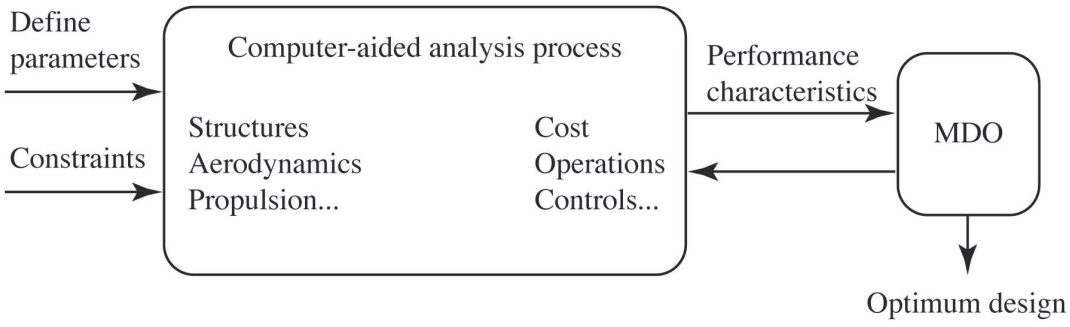


Figure A1. Computer-aided launch vehicle design.

RSM (ref. 25) is used to

- Obtain polynomial approximations to relationships between design objectives and design parameters in the form:

$$Y = b_0 + \sum b_i x_i + \sum\sum b_{ij} x_i x_j + \sum b_{ii} x_i^2 \qquad \text{(A1)}$$

- Capture individual effects, interactions between parameter and discipline, and nonlinearity (curvature)
- Determine optimum values and for sensitivity studies

Constructing the Response Surface Model

One way to construct the response surface model is to sample from the design space by using DOE techniques.

Fractional factorial DOE designs can be used to efficiently construct nonlinear approximation models.

- Central composite designs
- D-optimal designs for computer experiments

D-Optimal Designs

Most experimental designs are developed with the objective of minimizing experimental variance, so that outputs have neither measurement error nor variability. This objective is not always a good choice when approximating computer codes. D-optimal designs are typically considered a good approach for approximation model building using computer codes for irregularly shaped response models.

Steps in study process:

1. Conduct a screening study to efficiently select design parameters important to the system and the quality characteristics to be optimized.
2. Select parameter values (range) to bound the optimum.
3. Select appropriate orthogonal array (OA), i.e., the combination of experiments that will sample the design space.
4. Define the process and conduct analyses at points defined by the OA.
5. Construct math models (second-order, to capture nonlinearity).
6. Analyze data and models to obtain sensitivities.
7. Conduct multidisciplinary optimization to determine optimum levels for parameters.
8. Optimum may be an experiment not yet run, so conduct additional experiment to verify predicted parameters.

An example of applying D-optimal designs to a single-stage vehicle (SSV) follows.

Example 1: SSV Dual-Fuel Propulsion System Study

The SSV dual-fuel propulsion system study (refs. 10, 26, 27, 28) examined the proposed RD-701 Russian engines capable of burning hydrogen fuel alone, or hydrogen and hydrocarbon fuels simultaneously. The objective is to determine a combination of engine and vehicle design variable values to obtain an optimum (minimum empty weight) SSV design subject to constraints.

Steps 1 and 2: Identify Design Variables and Range of Values

Four propulsion system design variables with significant effects on SSV weight were studied. These parameters and their ranges are presented in table A1.

Table A1. Design Parameters and Ranges

Parameter	Range
Liftoff T/W_0	1.2–1.4
Transition Mach number	4–12
LH2 flow rate fraction	3–6
Nozzle expansion ratio	40–100

The objective is to determine a combination of design variable values to obtain an optimum weight SSV vehicle subject to constraints.

Step 3: Construct D-Optimal Design Matrix

A saturated (D-optimal) design matrix was constructed (ref. 29). Using this saturated design, we can study 4 variables at 4 values (levels) with 15 different point designs.

- 265 (4^4) possible design configurations reduced to 15
- Similar CCD study required 25 design points

Step 4: Define Process and Conduct Analyses

For this study, the following analyses were performed: trajectory analyses performed with the Program to Optimize Simulated Trajectories (POST); weights and sizing analyses performed with Configuration Sizing (CONSIZ) weights/sizing code; geometry and subsystem packaging analyses performed with Solid Modeling Aerospace Research Tool (SMART); and aerodynamic analyses performed with Aerodynamic Preliminary Analysis System (APAS). This procedure was repeated for each row of the design matrix, generating a vehicle design with a corresponding empty weight as shown in table A2(a). The values of the variables are shown in coded form in A2(a) and are defined in terms of actual values in table A2(b).

Table A2(a). Experimental Matrix and Results

	AR	T/Wo	LH2	Mach number	Empty weight, lb
1	–2	–2	–2	–2	197,100
2	2	–2	–2	–2	178,100
3	–2	2	–2	–2	185,200
4	–2	–2	2	–2	188,800
5	–2	–2	–2	2	166,100
6	0.82	0.82	–2	–2	180,500
7	0.82	–2	0.82	–2	179,100
8	0.82	–2	–2	0.82	162,400
9	–2	0.82	0.82	–2	187,900
10	–2	0.82	–2	0.82	170,700
11	–2	–2	0.82	0.82	165,400
12	–1.30	–2	2	2	173,400
13	2	–1.30	2	2	166,500
14	2	2	–1.30	2	184,400
15	2	2	2	–1.30	182,100

Table A2(b). Coded Values and Associated Actual Values

Coded	Actual test values			
	AR	T/W_0	LH2	Mach number
−2	40	1.20	3	4
−1.30	50	1.24	3.5	5.4
0.82	82	1.34	5.1	9.6
2	100	1.40	6	12

Steps 5 and 6: Construct Second-Order Model by Using Regression Analysis

Regression analyses were used to construct a second-order weight approximation model in terms of the four design variables. This model accounts for the curvature (square terms) of the response surface and the two-variable interactions (cross terms).

Empty weight =

166.73 + 0.129*(AR) + 2.32*(T/Wo) + 0.09*(LH2) - 3.03*(Mch) + 0.18*(AR)*(T/Wo) – 0.35*(AR)*(LH2) + 0.32*(AR)*(Mch) – 0.20*(T/Wo)*(LH2) + 0.42*(T/Wo)*(Mch)0.62*(LH2)*(Mch) +1.01*(AR)*(AR) + 0.16*(T/Wo)*(T/Wo) + 0.07*(LH2)*(LH2) + 1.87*(Mch)*(Mch)

Steps 7 and 8: Determine and Verify Parameter Values That Optimize Metric of Interest

The parameter values resulting from a minimization of the equation from steps 5 and 6 are given in table A3. Results obtained by using D-optimal design and CCD methods are shown in table A4. The optimized weight is significantly lighter than the reference vehicle and all 15 test configurations.

Table A3. Parameter Values Resulting from Optimization

Parameter	Range	Optimum value
Liftoff T/W_0	1.2–1.44	1.2
Transition Mach number	–12	10.7
LH2 flow rate fraction	3–6	6
Nozzle expansion ratio	40–100	73.8

Table A4. D-Optimal versus Central Composite Design

	D-Optimal	CCD
Number of design points	15	25
Predicted empty weight, lb	160,449	159,200
Verified empty weight, lb	161,100	161,000

Example 2: Single-Stage-to-Orbit, Wing-Body Configuration Study

The second example (refs. 30–32) involves a configuration optimization study of a rocket-powered, single-stage-to-orbit (SSTO) launch vehicle. The vehicle has a wing-body configuration with a slender, round cross section fuselage and a clipped delta wing. The delta wing has elevon control surfaces for aerodynamic roll and pitch control. The objective of the study was to determine a combination of design parameter values to obtain an optimum weight vehicle subject to aerodynamic constraints.

Steps 1 and 2: Identify Design Variables and Range of Values

Six weights and sizing and aerodynamic design parameters were varied over a fixed range. These parameters and their ranges are given in table A5.

Table A5. Design Parameters and Ranges

Parameter	Range
Fineness ratio (FR)	4–7
Wing area ratio (WA)	10–20
Tip fin area ratio (TFA)	0.5–3
Body flap area ratio (BFL)	0–1
Ballast weight (BL)	0–0.04
Mass ratio (MR)	7.75–8.25

Step 3: Construct D-Optimal Design Matrix

A saturated (D-optimal) design matrix was constructed (table A6). Using a saturated design, 6 parameters can be studied at 3 values with 45 point designs; 729 (3^6) possible design configurations reduced to 45. In table A6, −1 represents the lower boundary for a parameter, +1 represents the upper boundary, and 0 represents the midpoint between the two.

Table A6. D-Optimal Design Matrix

	FR	WA	TFA	BFL	BL	MR
1	–1	–1	–1	–1	–1	1
2	–1	–1	–1	–1	1	–1
3	–1	–1	–1	1	–1	0
4	–1	–1	–1	1	1	1
5	–1	–1	0	1	–1	1
6	–1	–1	1	–1	–1	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
41	1	1	0	0	–1	1
42	1	1	1	–1	–1	–1
43	1	1	1	–1	1	1
44	1	1	1	1	0	–1
45	1	1	1	1	1	1

Step 4: Define Process and Conduct Analyses

All geometry and subsystem packaging were performed by using SMART; APAS was used to determine vehicle aerodynamics, which were generated for Mach number = 0.3, 2, and 10. Weights and sizing analyses were performed by using the CONSIZ tool.

The process was repeated for the 45 rows of the D-optimal design matrix; each row corresponds to a vehicle design generating the weights and aerodynamic data. For the weights analysis, 45 data points for empty weight, landed weight, and landing centers of gravity were obtained. For the aerodynamics analysis, lift drag and pitching moment coefficients were obtained for the three Mach numbers. The experimental matrix is given in table A7.

Table A7. Experimental Matrix

	FR	WA	TFA	BFL	BL	MR	Weight	cg
1	–1	–1	–1	–1	–1	1	268,475	0.745
2	–1	–1	–1	–1	1	–1	264,659	0.716
3	–1	–1	–1	1	–1	0	260,610	0.745
4	–1	–1	–1	1	1	1	322,372	0.730
5	–1	–1	0	1	–1	–1	268,512	0.755
6	–1	–1	1	–1	–1	1	333,966	0.773
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
41	1	1	0	0	–1	1	261,486	0.756
42	1	1	1	–1	–1	–1	235,422	0.753
43	1	1	1	–1	1	1	318,675	0.741
44	1	1	1	1	0	–1	256,535	0.745
45	1	1	1	1	1	1	326,144	0.743

Step 5 and 6: Construct Second-Order Model by Using Regression Analysis

Regression analysis was used to determine the model coefficients for the weights and the aerodynamic data in terms of the six design parameters.

Empty weight =

271521 - 64701* FR + 41089* WA + 23540* TFA + 4014* BFL +
20011* BL + 22726 * MR + 30708 * FR * FR – 16300 * FR * WA + 338 * WA * WA –
8589 * FR * TFA – 934 * WA * TFA + 1286 * TFA * TFA – 1004 * FR * BFL – 971* WA *
BFL + 482 * TFA * BFL - 3049* BFL * BFL – 4063* FR * BL + 2862 * WA * BL + 1473 *
TFA * BL + 99 * BFL * BL – 1578* BL * BL – 4760 * FR * MR + 3582 * WA * MR + 1643*
TFA * MR + 287* BFL * MR + 2399* BL * MR + 1656* MR * MR

Response equations for landing velocity, entry trim, and stability constraints were also obtained.

CM, Supersonic =

-0.0204 + 0.007143 * FR - 0.008615 * WA + 0.000042 * TFA - 0.001214 * BFA +
0.002326 * ALPHA - 0.018623 * DELEV - 0.001038 * FR * WA + 0.001739 * FR *
ALPHA + 0.00042 * FR * DELEV + 0.000145 * WA * BFA – 0.002667 * WA * ALPHA -
0.001253 * WA * DELEV + 0.000257 * TFA * BFA – 0.000271 * BFA * ALPHA -
0.000296 * ALPHA * DELEV + 0.001197 * WA * WA - 0.000857 * DELEV * DELEV +
0.033439 * CG + 0.002126 * CG * FR – 0.002387 * CG * WA + 0.000181* CG * BFA +
0.007432 * CG * ALPHA + 0.003089 * CG * DELEV + 0.000044 * ALPHA * ALPHA

Steps 7 and 8: Determine and Verify Parameter Values That Optimize Metric of Interest

Nonlinear optimization and mathematical programming techniques are used to determine the design parameter values (table A8) to minimize weight subject to aerodynamic constraints.

Minimize empty weight: $\left(Y = b_o + \sum b_i x_i + \sum b_{ii} x_i^2 + \sum \sum b_{ij} x_i x_j\right)$

Subject to constraints:

CM Subsonic CM Supersonic CM Hypersonic Landing speed
Cg P/L, In Cg P/L, Out CL Subsonic CL Hypersonic
CD Subsonic CD Hypersonic

Table A8. Optimal Design Parameter Values

Parameter	Optimum value
Fineness ratio	6.9
Wing area ratio	18.76
Tip fin area ratio	1.99
Body flap area ratio	0
Ballast weight	0.014
Mass ratio	8.0

- Predicted empty weight: 249,792 lb
- Verified empty weight: 249,360 lb

Empty weight optimized; significant reduction compared with reference vehicle; second-order approximation equations accurately represented response surface.

Conclusions

Response surface methods

- Offer a systematic, flexible approach for studying the parameter space efficiently
- Capture variable and discipline interactions
- Enable sensitivity simulations to be carried out rapidly
- Enable multidisciplinary integration of analysis codes through the response surface math (approximation) models
- Provide considerable benefits in terms of optimized designs and reduced engineering effort and resources

RSM using D-optimal designs

- Design points reduced to a minimum; second-order or cubic response surface polynomials can be constructed
- Efficient model building, multidisciplinary integration, and fast analysis with low expenditure of experimental effort
- Good approximation of the response surface
- Good approach for approximation model building using computer codes (in most cases)

References

1. Blair, J. C.; Ryan, R. S.; Schutzenhofer, L. A.; and Humphries, W. R.: *Launch Vehicle Design Process: Characterization, Technical Integration, and Lessons Learned*. NASA/TP-2001-210992, 2001.

2. Hammond, Walter E.: Space Transportation: A Systems Approach to Analysis and Design. AIAA Education Series, 1999.

3. Rowell, L. F.: Design Challenges for Tomorrow's Manned Launch Systems. AIAA-93-1068, Feb. 1993.

4. Dumbacher, D. L.: NASA 2nd Generation RLV Program Introduction, Status, and Future Plans. AIAA-2002-4570, July 2002.

5. Monroe, R. W.; Lepsch, R. A.; and Unal, R.: Risk Analysis of Weight Estimates for a Launch Vehicle. SAWE Paper No. 2323, June 1996.

6. Unal, R.; Morris, W. D.; White, N. H.; Lepsch, R. A.; and Brown, R. W.: Approximation Model Building for Reliability and Maintainability Characteristics of Reusable Launch Vehicles. AIAA-2000-4712, Sept. 2000.

7. Greenberg, Joel S.: *Space Transportation Architecture Comparisons Explicitly Considering Uncertainty and Risk*. Princeton Synergetics, 1996.

8. Moore, A. A.; Braun, R. D.; and Powell, R. W.: The Infusion of Cost Into the Multidisciplinary Design of Space Transportation Systems. *Proceedings of the 17th Annual Conference of the International Society of Parametric Analysts*, May 1995.

9. Alexandrov, N.; and Hussaini, Y., eds.: *Multidisciplinary Design Optimization: State of the Art.* SIAM Publ., 1997.

10. Rowell, L. F.; Braun, R. D.; Olds, J. R.; and Unal, R.: Multidisciplinary Conceptual Design Optimization of Space Transportation Systems. *J. Aircr.*, vol. 36, no. 1, 1999, pp. 218–226.

11. Wilson, J. W.; Korte, J. J.; Heinbockel, J. H.; Clowdsley, M. S.; Badavi, F. F.; and Troung, A.: Ionizing Radiation: Multifunctionality and MDO Processes. Paper Number 02ICES-336, ICES, 2002.

12. Carle, A.; and Fagan, M.: Overview of Adifor 3.0. CAAM-TR 00-02; Rice Univ., Jan. 2000.

13. Braun, R. D.; and Kroo, I. M.: Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment. *Multidisciplinary Design Optimization: State of the Art.* N. Alexandrov and Y. Hussaini, eds., SIAM Publ., 1997, pp. 98–116.

14. Chen, Wei; and Lewis, Kemper: A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design. AIAA-98-4945, Sept. 1998.

15. Newman, J. C., III; Whitfield, D. L.; and Anderson, W. K.: A Step-Size Independent Approach for Multidisciplinary Sensitivity Analysis and Design Optimization. AIAA-99-3101, June 1999.

16. Newman, J. C., III; Anderson, W. K.; and Whitfield, D. L.: Multidisciplinary Sensitivity Derivatives Using Complex Variables. MSSU-EIRS-ERC-98-08, Engineering Research Center Rep., Mississippi State Univ., July 1998.

17. Hammond, Walter E.: Design Methodologies for Space Transportation Systems. AIAA Education Series, 2001.

18. Olds, J. R.: Results of a Rocket-Based Combined-Cycle SSTO Design Using Parametric MDO Methods. SAE 94-1165, 1994.

19. Engelund, W. C.; Stanley, D. O.; Lepsch, R. A.; McMillin, M.; and Unal, R.: Aerodynamic Configuration Design Using Response Surface Methodology Analysis. AIAA-93-3967, Aug. 1993.

20. Braun, R. D.; Powell, R. W.; Lepsch, R. A.; Stanley, D. O.; and Kroo, I. M.: Comparison of Two Multidisciplinary Optimization Strategies for Launch Vehicle Design. *J. Spacecr. & Rockets*, vol. 32, no. 3, 1995, pp. 404–410.

21. Unal, R.; Braun, R. D.; Moore, A. A.; and Lepsch, R. A.: Design Optimization for Cost Using Genetic Algorithms. *Proceedings of the 17th Annual Conference of the International Society of Parametric Analysts*, May 1995, pp. 183–191.

22. Unal, R.; Braun, R. D.; Moore, A. A.; and Lepsch, R. A.: Design Optimization on Cost Basis Using Taguchi's Orthogonal Arrays. *Proceedings of the American Society for Engineering Management National Conference*, ASEM, Oct. 1996, pp. 17–21.

23. Moore, A. A.; Braun, R. D.; Powell, R. W.; and Qualls, G. D.: Determination of Optimal Launch Vehicle Technology Investment Strategies During Conceptual Design. AIAA 96-4091, Sept. 1996.

24. Putko, M. M.; Newman, P. A.; Taylor, A. C., III; and Green, L. L.: Approach for Uncertainty Propagation and Robust Design in CFD Using Sensitivity Derivatives. AIAA 2001-2528, June 2001.

25. Myers, R. H.: *Response Surface Methodology*. Allyn and Bacon, 1971.

26. Lepsch, R. A., Jr.; Stanley, D. O.; and Unal, R.: Dual-Fuel Propulsion in Single Stage Advanced Manned Launch System Vehicle. *J. Spacecr. & Rockets*, vol. 32, no. 3, 1995, pp. 417–425.

27. Lepsch, R. A.; Stanley, D. O.; and Unal, R.: Application of Dual-Fuel Propulsion to a Single Stage AMLS Vehicle. AIAA-93-2275, June 1993.

28. Unal, R.; Lepsch, R. A.; Engelund, W. C.; and Stanley, D. O.: Approximation Model Building and Multidisciplinary Design Optimization Using Response Surface Methods. AIAA-96-4044, Sept. 1996.

29. Box, G. E.; and Draper, N. R.: *Empirical Model Building and Response Surfaces*. John Wiley, 1987.

30. Unal, R.; Lepsch, R. A.; and McMillin, M.: Response Surface Model Building and Multidisciplinary Optimization Using D-Optimal Designs. AIAA-98-4759, Sept. 1998.

31. Stanley, D. O.; Engelund, W. C.; Lepsch, R. A.; McMillin, M. M.; Wurster, K. E.; Powell, R. W.; Guinta, A. A.; and Unal, R.: Rocket-Powered Single-Stage Vehicle Configuration Selection and Design. *J. Spacecr. & Rockets*, vol. 31, no. 5, 1994.

32. Engelund, W. C.; Stanley, D. O.; Lepsch, R. A.; McMillin, M.; and Unal, R.: Aerodynamic Configuration Design Using Response Surface Methodology Analysis, AIAA-93-3967, Aug. 1993.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 10 - 2003	Technical Memorandum	

4. TITLE AND SUBTITLE

Launch Vehicle Design and Optimization Methods and Priority for the Advanced Engineering Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Lawrence F. Rowell and John J. Korte

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

23-721-10-41

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-18337

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2003-212654

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 15
Availability: NASA CASI (301) 621-0390 Distribution: Nonstandard

13. SUPPLEMENTARY NOTES

An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov

14. ABSTRACT

NASA's Advanced Engineering Environment (AEE) is a research and development program that will improve collaboration among design engineers for launch vehicle conceptual design and provide the infrastructure (methods and framework) necessary to enable that environment. In this paper, three major technical challenges facing the AEE program are identified, and three specific design problems are selected to demonstrate how advanced methods can improve current design activities. References are made to studies that demonstrate these design problems and methods, and these studies will provide the detailed information and check cases to support incorporation of these methods into the AEE. This paper provides background and terminology for discussing the launch vehicle conceptual design problem so that the diverse AEE user community can participate in prioritizing the AEE development effort.

15. SUBJECT TERMS

launch vehicle, conceptual design, collaboration, environment, response surface, design of experiments

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	35	(301) 621-0390